



Práctica 6:

TABLAS HASH

OBJETIVO

- Implementación de tablas hash en C++.
- Comparación de ejecución entre tablas hash y árboles binarios.

MATERIAL A ENTREGAR

- *.h, *.cpp
- Makefile
- coste.png
- Ejercicios apartado 1. (previo)

DESARROLLO

1. Ejercicios previos

Escribe el código para la función de hash que vas a utilizar.

2. Clase Tabla

Implementar la clase Tabla a partir de tabla.h y las transparencias de clase. La clave deberá ser un string, el dato a guardar será un objeto de tipo *Alumno*. Utilizar *alumno.h* y *alumno.cpp* de la práctica pasada.

3. Prueba de la clase Tabla

Para comprobar el buen funcionamiento de la clase Tabla, se deberá escribir un programa (**test_tabla.cpp**) que construya una tabla hash que contenga 1.000 alumnos. Después se deberán realizar los siguientes pasos:

1. Mostrar la tabla hash en pantalla
2. Buscar 3 DNI. 1 de ellos no debe existir. Si el código existe, mostrar el alumno por pantalla.
3. Mostrar el **factor de carga** de la tabla y el número de posiciones de la tabla que no están usadas.

¿Están los elementos correctamente distribuidos en la tabla?

4. Costes

Implementa el programa **costes.cpp**, que deberá calcular el **tiempo medio** (en segundos) empleado en buscar (con éxito) **un** alumno almacenado en las 3 estructuras. Se deben calcular los costes para tallas del problema (número de alumnos leídos) de 500 a 5000 en incrementos de 500. Para ello se implementará una función *calculoCoste* similar a la práctica 4:

```
void calculoCoste(unsigned ini, unsigned fin, int inc, float (*f)(unsigned),  
ostream & file);
```



Se deberá implementar una función auxiliar para cada estructura. Dentro de esta función auxiliar crearemos en primer lugar la estructura (en el caso de la tabla, de tamaño doble de la talla). A continuación insertaremos **n alumnos** (donde n es la talla) en la estructura. A la vez que se introducen los alumnos en las estructuras, se deberán guardar los DNI en un vector de strings que se utilizará para realizar las búsquedas. Por último **buscaremos los n alumnos** en la estructura utilizando el vector de DNI. Esta búsqueda se deberá repetir un número suficiente de veces (mínimo 100) para poder medir el tiempo correctamente.

En la función calculoCoste se deberá dividir el tiempo por el valor adecuado para obtener el **coste medio de una búsqueda**.

Representar todos los datos obtenidos en una única gráfica *costes.png*.

¿Qué conclusiones obtienes de las gráficas?